IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

APPEAL NO:

In Re Application of: Pagan, et al.

Confirmation No. 9632

Serial No.: 10/602,425

Filed: June 24, 2003

Title: METHOD AND SYSTEM FOR PROVIDING INTEGRATED HOT KEY CONFIGURATION

REPLY BRIEF ON APPEAL

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:	Date: January 14, 2009
William G. Pagan, et al.	Confirmation No. 9632
Serial No.: 10/602,425	Group Art Unit: 2175
Filed: June 24, 2003	Examiner: Nunez, Jordany

For: METHOD AND SYSTEM FOR PROVIDING INTEGRATED HOT KEY CONFIGURATION

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF ON APPEAL

Pursuant to 37 CFR 41.41(a)(1), Appellant responds to the points raised in the Examiner's Answer mailed November 14, 2008, as follows:

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the Appeal Brief.

(2) Related Appeals and Interferences

A statement identifying the related appeals and interferences is contained in the Appeal Brief.

(3) Status of Claims

A statement identifying the status of the claims is contained in the Appeal Brief.

(4) Status of Amendments

A statement identifying the status of amendments is contained in the Appeal Brief.

(5) Summary of Claimed Subject Matter

A summary of the claimed subject matter is contained in the Appeal Brief.

(6) Grounds of Rejection to be Reviewed on Appeal

A statement identifying the grounds of rejection to be reviewed on appeal is contained in the Appeal Brief.

(7) Response to Examiner's Answer

1. Examiner's Answer, Page 11, Section (10), Regarding the 35 USC 102(e) rejection of Claims 1-4, 12-15, 23-26, 35, 39, 43, 49

The Examiner stated in Section 10, Part 1), "Slaunwhite (page 2, paragraph [0035]) teaches that there could be several instances of the same item type displayed simultaneously. Further, Slaunwhite teaches (page 3, paragraph [0037]) that a user typically (e.g., not always) selects an item type from a customization dialog containing a list of available item types and then keys in the shortcut key that is associated with it. Then, as one of ordinary skill in the art would understand, different item instances of the same item type are displayed both in the toolbar and the customization dialog. Thus, Slaunwhite clearly teaches a particular function (e.g., task) being performed when a corresponding displayed item is selected by the pointing device (e.g., performing a task by selecting an item type on a toolbar), and the particular function (e.g., task associated with the item type) of the displayed item being indicated for mapping when the pointing

device is moved over that displayed item (e.g., selecting the particular function for mapping a shortcut when the item type is selected from a list on a customization dialog)."

However, Slaunwhite does not disclose that different item instances of the same item type are displayed in both the toolbar and the customization dialog. Slaunwhite teaches that an "instance" can be a command user interface item that provides a command/request (paragraphs [0031, 0033]), or an instance can be a non-command user interface item that changes an internal setting (paragraphs [0032, 0034]). In contrast, the customization dialog is described to provide a list of "available item types," not instances. Nowhere is it stated in Slaunwhite that the "item types" in the customization dialog are instances. The instances provide a command/request or change an internal setting, while the item types in the customization dialog are simply a description of a type, and provide that type. Indeed, Slaunwhite states that "the item type is a definition of an item and does not represent an actual instance of an item but uniquely defines a type of an item" (paragraph [0029]). Thus Slaunwhite cannot disclose that different item instances of the same item type are displayed in both the toolbar and the customization dialog.

Furthermore, even if a type listed in the customization dialog is considered to be an instance (for the sake of argument), Slaunwhite would only disclose or suggest displaying one instance on a toolbar used for performing a task, and displaying a different, other instance in a customization dialog used for mapping a shortcut. Slaunwhite nowhere suggests allowing a function to be performed using a displayed item and indicating that function for mapping using that same displayed item, as recited in claim 1 (where "item" is the closest analogy to Slaunwhite's "instance," as is evident

throughout Appellant's specification, e.g., pages 5 and 8, as well as Slaunwhite's use of "item", e.g. paragraph [0033]).

The Examiner also stated that "Slaunwhite teaches (page 2, paragraph [0036]) that when defining a shortcut key assignment, an item receiver (e.g., program module) receives the item type from a user input unit (e.g., by selecting item type with a mouse), without mentioning any customization dialog. As stated above, Slaunwhite (page 2, paragraphs [0030] [0035]) teaches that an item type instance may be displayed on a toolbar and that an item type instance may selected using a mouse. In other words, Slaunwhite teaches that an item receiver may receive the item type by selecting an item type instance from a toolbar with a mouse (e.g., user input unit). Thus, one of ordinary skill in the art would readily understand that Slaunwhite teaches at least two ways of selecting the particular function for mapping a shortcut: a) typically by selecting item type instance a list on a customization dialog and b) when an item type instance is selected from a toolbar."

However, one of ordinary skill would not readily understand from Slaunwhite that mapping a shortcut can be performed when an item type instance is selected from a toolbar. The fact that Slaunwhite does not mention any customization dialog when receiving an item type for mapping is not an indication that any toolbar icon or instance can be used for this purpose. Slaunwhite simply provides a broad, general description in paragraph [0036] of receiving an item type for mapping to a shortcut, and that broad description is further detailed in the following paragraph [0037], where the use of a customization dialog is taught to send the item type for mapping. The fact remains that the only example that Slaunwhite describes for inputting a type for mapping to a shortcut

is the use of a list of types in a customization dialog. A broad description of receiving a type for mapping, followed by a single example of using a customization dialog to implement the sending for this reception, in no way suggests that a toolbar instance/item can be used for mapping. Slaunwhite's described other uses of displayed items / instances are nowhere suggested to be related to mapping shortcuts. For example, Slaunwhite teaches that "the user can use the item to perform a specific task" (paragraph [0030]) and, as explained above, implies that displayed items can be selected to provide a command/request (paragraph [0031]) or to change an internal setting (paragraph [0032]), but nowhere suggests that such items / instances are selected to send a type for mapping.

Furthermore, as explained above, nowhere is it mentioned or suggested in Slaunwhite that the same displayed item can be used with a pointer for both performing a function and indicating the function for mapping to a shortcut, as recited in claim 1.

The Examiner made similar points in Parts 4), 5), 6), 7) and 9) for these and other claims, which are addressed similarly as the explanation above.

The Examiner stated in Section 10, part 2), that "Appellant's statement 'the items displayed in Slaunwhite's mapping customization dialog are nowhere taught to be selectable to perform a corresponding function' is incorrect, at least because it contradicts another of Appellant's statement 'Slaunwhite teaches only one way for the item receiver 102 to receive the item type from the user input unit 10, in paragraph [0037] a customization dialog where the user selects the item type from a list of available item types and then keys in the shortcut key that is associated with it.' In other words, the items displayed in Slaunwhite's customization dialog are 'selectable to perform a

corresponding function' of being received by the item receiver in order to define a shortcut key assignment."

However, Appellant's statement of a "corresponding function" here referred to the "particular function" in the context of claim 1. For example, as recited in claim 1, "the mapping causes the particular function to be accessed by the computer system when the mapped hot key is selected." The sending of an item type listed in Slaunwhite's customization dialog for mapping is not itself the function mapped to the hot key, since the command item type or non-command item type described or represented by the listed type is what is mapped to the hot key (not the sending of that item for mapping).

The Examiner made similar points in part 3), which are addressed similarly as the explanation above.

2. Examiner's Answer, Section (10), Regarding the rejection of Claims 37, 41, and 45

The Examiner stated in Section 10, part 8), that "Slaunwhite teaches (as shown in figure 5) that when a user sets focus to a zoom level interface type, a portion of the text of the item indicating the zoom level is selected. Because Slaunwhite shows this being part of selecting a user interface item type, one of ordinary skill in the art would readily understand that when user sets focus to a zoom level interface type, thereby selecting a portion of the text of the item indicating the zoom level, the zoom level is indeed being used to indicate a function for shortcut key mapping."

However, setting focus to the zoom level interface as described in Slaunwhite is nowhere related to Slaunwhite's input for mapping to a hotkey. Fig. 5 illustrates a zoom

popup window displayed after the user has selected a shortcut key mapped to the zoom window. Focus is set to the zoom level text in the zoom window so that the user can modify the text with the keyboard. This is not related to indicating the zoom function for mapping the zoom function to a hotkey; the hot key has already been mapped (to the Alt-Z hotkey, paragraph [0051]).

Even if (for the sake of argument) setting focus on this zoom window text is considered selecting a type, nowhere does Slaunwhite disclose or suggest using a same displayed text item with a pointer for both performing a function and indicating the function for mapping to a shortcut, similarly as explained previously.

3. Examiner's Answer, Section (10), Regarding the rejection of Claims 46, 47, and 48

The Examiner stated in Section 10, part 10), that "one would have been motivated to make such combination because a way to simplify the way in which a user accesses a particular non-command user interface item would have been obtained and desired, as expressly taught by Slaunwhite (page 1, paragraph [0010])... absent evidence that the modifications necessary to effect the combination of elements is uniquely challenging or difficult for one of ordinary skill in the art, the claim is unpatentable as obvious..."

However, Slaunwhite's use of an instance to perform of a task and use of a different list of types to map a hotkey does not obviously lead to using one instance or item to perform both of these abilities. Slaunwhite completely fails to disclose or suggest in detail anything but a standard way to select types for mapping to hotkeys (using a separate list of types in a customization dialog) and so nowhere suggests using the same

displayed item (i.e., instance) for both performing a function and indicating a function for mapping to a hot key, despite the benefits to doing so (such as not having to look down a list for a specific type to map). Furthermore, assigning a selected letter of the text item as the mapped hotkey is nowhere disclosed or suggested in any of the cited references. For example, Slaunwhite only discloses the user "keying in" a shortcut key, not selecting a portion of a displayed item to become the shortcut key, nor using the same displayed item for both performing and indicating the hotkey in such a manner. Appellant therefore believes that the claimed subject matter would not be obvious in view of Slaunwhite.

4. Examiner's Answer, Section (10), Regarding the rejection of Claims 10, 21, 32, and 36

The Examiner stated in Section 10, part 11), "Appellant, by failing to argue otherwise, seemly acknowledges that clicking and hovering are indeed obvious variants of each other when trying to indicate an action. Further, Appellant, seems to try and make the case the in Appellant's application the clicking performs one action while the hovering performs a different action, and that this would not have been known to one of ordinary skill in the art. This is wrong because: 1) Slaunwhite teaches that clicking over the same displayed item, as stated above, can be used to perform two different actions, for example a) (page 2, paragraph [0030]) performing a task and b) (page 2, paragraph [0037]) select a type instance from a list on a customization dialog), and thus clicking and hovering, being obvious variants of each other, could clearly have been used to perform the two different actions; 2) one of ordinary skill in the art would have been well aware that clicking on an toolbar item might perform a task associated with that toolbar item,

while hovering over that same toolbar item might perform the second action of providing a user with information regarding that toolbar item. Thus, … one of ordinary skill in the art could have combined the elements as claimed by known methods (e.g., as taught by Slaunwhite and Forest), and in combination, each element merely would have performed the same function as it did separately."

However, clicking and hovering are not obvious variants when each is allowed to perform a different action on the same displayed item, those different actions being performing a particular function, and indicating that function for mapping to a hot key, similar to the explanations for the claims above. Furthermore, as explained previously, nowhere does Slaunwhite disclose or suggest that "clicking over the same displayed item" can be used to perform two different actions; the item mentioned in paragraph [0030] for performing a task is an instance that is different than the type provided in the customization dialog for mapping. Furthermore, hovering over a toolbar item to provide a user with information is not the same as indicating a function for mapping, nor would indication for mapping be obvious in view of such hovering. This is shown at least by Slaunwhite and Forest failing completely to mention or suggest mapping hotkeys using anything but a standard separate list of types, despite the benefits of other selection methods such as claimed by Appellant.

The Examiner made similar points in part 12), which are addressed similarly as the explanation above.

Conclusion

Slaunwhite and Forest do not disclose or suggest the claimed inventions as argued above and in the Appeal Brief. Appellant, therefore, respectfully submits that the pending claims are not properly rejected under §§ 102 and 103.

For these reasons, and the reasons stated in the Appeal Brief, Appellant submits that the final rejection should be reversed.

Please apply any charges or credits to Deposit Account No. **50-0563**.

Respectfully submitted,
SAWYER LAW GROUP LLP

January 14, 2009

/Joseph A. Sawyer, Jr./
Joseph A. Sawyer, Jr.
Attorney for Appellant
Reg. No. 30,801
(650) 493-4540

Customer Number 47052